MYCRO TRANSYT :
UN ENVIRONNEMENT MICRO-INFORMATIQUE POUR LA REGULATION DES FEUX DE CIRCULATION

par

Marc Kaltenbach*

* Bishop Univesity, Lennoxville (Québec)

Ce rapport a été publié grâce à une subvention du Fonds F.C.A.C. pour l'aide et le soutien à la recherche.

Université de Montréal
Centre de recherche sur les transports - Publication #349
Juillet 1984

Dépôt légal - 3e trimestre 1984

SOMMAIRE

Un logiciel destiné à faciliter la sélection des paramètres de la commande des feux de circulation en milieu urbain est décrit. Une restructuration et adaptation du programme Transyt est proposée visant à rendre ce programme plus performant, "user friendly", et utilisable sur micro-ordinateurs.

REMERCIEMENTS

Nous tenons à remercier le Fonds F.C.A.C. qui a subventionné cette recherche, le Centre de recherche sur les transports de l'Université Bishop's. Un crédit particulier revient à M. J.-P. Turgeon qui a élucidé pour nous certains aspects du programme Transyt.

Introduction

Transyt est un logiciel pour optimiser la commande des feux de circulation aux intersections d'un réseau de voies urbaines couramment utilisé dans plusieurs grandes villes du Royaume Uni et ailleurs dans le monde. Les paramètres de la commande que l'on cherche à optimiser à l'aide de ce programme sont la durée du cycle qui fixe la période des alternances de phases (rouges, vertes, oranges, ...etc) commune à toutes les intersections du réseau et, pour chaque intersection, le "split" et l' "offset". Le "split" détermine la fraction du cycle au cours de laquelle les véhicules sur une voie donnée ont le droit de passage et l' "offset" fixe le décalage dans le temps du début du cycle à l'intersection par rapport à un temps de référence fixe. Transyt détermine les valeurs de la commande aux intersections (noeuds) du réseau qui minimisent une somme pondérée des retards et arrêts subits par les véhicules emprumptant le réseau au cours d'une durée égale à un cycle, représentative des conditions de flot moyennes dans le réseau. Pour qu'il en soit ainsi, il faut que les arrivées de véhicules à chaque noeud du réseau se répètent identiquement d'un cycle au suivant, une situation à laquelle nous réferrons dans ce qui suit en disant que les distributions de flots sont stationnaires. Nous mentionnerons plus loin le problème l'existence de ces distributions stationnaires.

Le programme Transyt dans sa version la plus récente (version 8) est le résultat d'une évolution lente à partir d'un programme initial conçu par D.I. Robertson et une équipe de chercheurs du TRRL (1),(2), en Angleterre au milieu des années 60. Le programme actuel est massif, ayant plus de 5000 lignes codées en fortran qui révèlent très peu de la logique sous-jacente au programme. Il utilise un espace de mémoire considérable et il faut le segmenter pour pouvoir le mettre en place même sur une grosse unité de calcul (Cyber 170 de l'Université de Montréal). De surcroît, le manuel d'utilisation et les articles disponibles dans la litérature sont fort peu explicites quant au détail des calculs effectués.

L'objectif du travail rapporté ici a été de repenser Transyt afin d'en obtenir une forme mieux structurée, plus compacte et facile à utiliser tant pour des fins de recherche que pour des fins opérationnelles. Ceci nous a conduit à formuler un ensemble de procédures en langage pascal constituant un environnement interactif plus flexible que Transyt et compatible avec une utilisation sur micro-ordinateurs. Par environnement, nous entendons que l'utilisateur dispose d'un choix de modules pour l'acquisition des données, l'optimisation de la commande, la visualisation du suivi des calculs effectués et des résultats, qui peuvent être facilement assemblées en une variété très grande de programmes pour des études et résultats spécifiques. Dans cet article nous décrivons les principaux aspects de ce logiciel appelé Micro-Transyt pour le distinguer de Transyt. Cette exposition procure aussi le contexte pour proposer des améliorations et variantes à Transyt, tant au point de vue conceptuel qu'au point de vue de l'utilisation par des utilisateurs non-spécialistes. La structure de cet article suit les principales étapes d'une utilisation de Micro-Transyt.

I - Acquisition des données

Le dessin de procédures interactives d'acquisition des données a pris en compte trois critères principaux qui sont,

1) de minimiser l'effort d'entrée des données grâce à l'utilisation de règles d'inférence pour générer les données redondantes ou éliminées par des données déjà entrées,

2) d'assurer la cohérence des données,

3) de faciliter les modifications des données et analyses de sensibilité de la performance par rapport à certains paramètres des modèles utilisés.

Les données, groupées en quatre catégories principales, sont acquises dans l'ordre suivant,

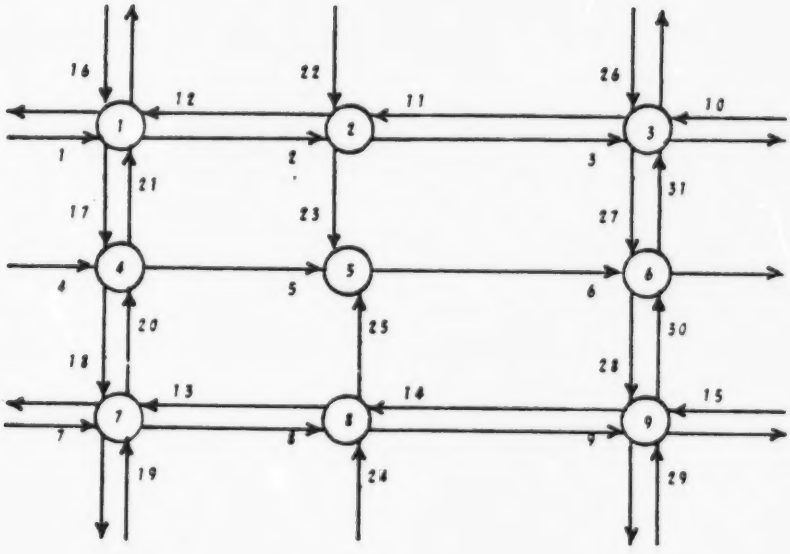

Figure 1 : Réseau de 9 noeuds et 31 arcs.
Les possibilités de tourner aux noeuds
ne sont pas indiquées.

a) La topologie du réseau

La figure 1 représente un réseau de rues et leurs intersections, typique pour une utilisation de Micro-Transyt, à part la taille du réseau, mesurée par le nombre de voies de circulation (arcs) entre intersections (noeuds) et le nombre de noeuds, qui peut être beaucoup plus grande. Sur la figure, les arcs et les noeuds sont numérotés dans un ordre arbitraire avec des entiers naturels. Pour chaque arc I du réseau il est nécessaire d'entrer la longueur de l'arc, la vitesse moyenne V_I des véhicules en parcours libre sur l'arc, le noeud N = EndNode(I) marquant la sortie de l'arc I, les arcs avals NextLink(I,d) selon trois directions possibles, d ϵ(tourne à gauche, tout droit, tourne à droite). On entre les valeurs Nextlink(I,d) = 0 si l'arc aval selon la direction d existe mais ne fait pas partie du réseau et NextLink(I,d) = -1 si la direction d à la sortie de I n'est pas permise (ou n'existe pas). On fournit également le taux maximum Serv(I) avec lequel les véhicules peuvent sortir de l'arc quand ils ont le droit de passage et les fractions FracTrac(I,d) selon lesquelles les véhicules se divisent parmi les directions de sortie possibles.

b) Le pas de discrétisation

Les modèles de simulation utilisés pour évaluer la performance sont discrétisés dans le temps par le choix du pas de simulation Del (par exemple Del = 5 sec.). Autrement dit, les modèles proposés décrivent les flots de véhicules dans le réseau toutes les Del secondes. Ceci simplifie considérablement la structure des modèles au prix d'approximations mineures sur les longueurs des arcs et le choix des instants auxquels il est possible de changer les indications des feux de circulation. Ainsi la longueur de l'arc I est approximée par un multiple de l'intervale de longueur V_I * Del. Il résulte que l'on peut représenter les flots de véhicules sur l'arc I aux instants Del, 2*Del, ...etc, par une fonction constante par morceaux de longueur V_I * Del. La hauteur d'un bloc de l'histogramme résultant, sur la figure (2), indique un taux de passage (véh./sec.) de véhicules en chaque

point du bloc. A chaque pas de simulation, l'histogramme est translaté d'un bloc dans la direction du flot. Le choix d'une valeur moindre pour Del améliore l' approximation de la distribution réelle des véhicules sur l'arc I mais accroît très rapidement la taille des calculs et l'espace de mémoire nécessaires pour identifier une commande optimale. Il est toutefois possible de réduire la valeur de Del au cours de l'optimisation pour rendre les modèles plus exacts dans le voisinage d'un optimum.

c) La commande aux noeuds

L'optimisation de la commande se fait par améliorations successives jusqu'à ce qu'il ne soit plus possible d' améliorer la valeur d'un seul paramètre de commande indépendamment des autres paramètres. Il faut donc proposer une commande initiale. Cette commande est sous la forme de valeurs entières qui, lorsque multipliées par Del, donnent des durées. Ce sont le paramètre CYCLE commun à tous les noeuds du réseau et, pour chaque noeud N, les valeurs SPLIT(N) et OFFSET(N). La donnée de ces valeurs permet de définir une fonction Rgt(I,Step) qui détermine si les véhicules à l'extrémité de sortie de l'arc I ont le droit de passage pendant l'intervale (Step*Del,(Step+1)*Del). Rgt peut également prendre en compte les phases intermédiaires (orange, vert avancé, ...etc.) et les retards d'accélération de véhicules en début de vert.

d) Les distributions initiales ou moyennes de flots sur les arcs.

Dans le cas de Transyt on spécifie des taux constants d'arrivées de véhicules aux entrées du réseau. Micro-Transyt accepte des distributions périodiques de période égale à un cycle (ce qui inclut le cas de Transyt comme cas particulier). Cette extension facilite la décomposition d'un grand réseau en sous-réseaux de tailles compatibles avec les limites du logiciel. Par exemple le nombre maximum de noeuds accepté par Transyt-8 est de 50 noeuds, si bien qu'en général lorsque Transyt est appliqué à une grande ville les arrivées de flots aux entrées du réseau sont elles aussi controllées

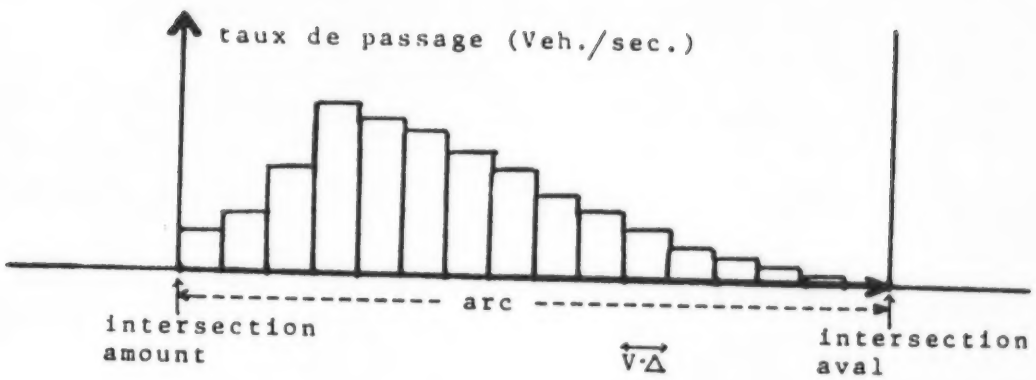

Figure 2 : Distribution des Véhicules en mouvement libre sur un arc.

par des feux de circulation et donc périodiques. Si les arrivées aux entrées du réseau n'ont pas la période du cycle, le modèle souvent n'admettra pas de distributions stationnaires pour les autres arcs du réseau et il faudra simuler l'évolution des flots sur plusieurs cycles pour évaluer correctement la performance de la commande, ce qui pourra entrainer un coût de calcul prohibitif.

Transyt requiert de l'utilisateur qu'il spécifie le volume moyen des flots pour chaque arc du réseau. Au cours des calculs effectués par Transyt les valeurs des flots sur les arcs sont constamment renormlisées pour être en accord avec ces volumes moyens. Notre modèle laisse à l'utilisateur le choix de ne pas imposer ces spécifications et d'obtenir les volumes moyens résultant du choix des coefficients de tourne aux noeuds. Il peut ainsi par ajustements successifs des coefficients de tourne obtenir une bonne concordance entre le modèle et certains volumes moyens de flots mesurés dans la réalité entre origines et destinations dans le réseau. Il est à remarquer que, dans les cas de Transyt et de Micro-Transyt, les volumes moyens de flots sur les arcs ne sont pas affectés par les changements de la commande aux noeuds. Finalement, comme pour Transyt, on peut spécifier une valeur pour le lissage exponentiel des distributions d'arrivées de flots sur les arcs afin d'approximer le phénomène d'étalement des pelotons de véhicules en mouvement. L'existence de distributions stationnaires ne semble pas être affectée par ce lissage.

II - Simulation des flots de véhicules

Les modèles de simulation de flots de véhicules dans un réseau sont utilisés pour calculer les distributions stationnaires d'arrivées de véhicules aux intersections du réseau et pour déterminer la valeur de la performance associée avec une commande particulière aux noeuds. Deux modèles de simulation, identiques quant aux résultats d'une simulation, mais très différents quant aux espaces de mémoire et aux temps de calculs qu'ils requièrent, sont inclus dans Micro-Transyt.

1) Simulation globale

Un modèle de simulation s'appliquant à l'ensemble du réseau étudié consomme beaucoup d'espace de mémoire. Ceci conduit à modifier la formulation du modèle dans Transyt pour la rendre conforme à celle d'un système dynamique d'état. Ainsi, à chaque instant Del, 2*Del, ..., CYCLE * Del,...etc. de la simulation, on ne garde en mémoire qu'un vecteur d'état de taille minimale, c.à.d. composé seulement des valeurs des flots libres sur chaque arc du réseau (hauteurs des blocs pour les histogrammes représentés par la figure (2) et du nombre de véhicules à l'arrêt (formant des queues) sur les arcs. Le modèle de simulation exprime les changements dans le vecteur d'état entre les instants successifs Del,...,CYCLE * Del,...etc. Une description détaillée de ce modèle est contenue dans la référence (3). Pour obtenir des distributions stationnaires pour tous les arcs du réseau on se contentera de poursuivre la simulation un pas de simulation après l'autre jusqu'à ce que l'indice de performance calculé pour les derniers CYCLE pas de simulation soit égal, avec une tolérance epsilon, à l'indice calculé pour les CYCLE pas précédents.

2) Simulation locale

Micro-Transyt est un logiciel consacré à l'optimisation des paramètres de commande des feux de circulation. La procédure d'optimisation, similaire à celle de Transyt, suit une forme de relaxation ("hill-climbing") qui cherche à améliorer la valeur d'un seul paramètre de la commande à la fois, les autres étant fixés à leurs meilleures valeurs trouvées jusqu'à présent. L'optimisation termine lorsqu'il n'est plus possible d'améliorer la valeur d'un seul paramètre indépendamment des autres. Chaque changement de la valeur d'un paramètre nécessite le recalcul des distributions stationnaires de flots. Pour obtenir ces distributions il peut être nécessaire de simuler l'évolution des flots dans le réseau pour une durée de plusieurs cycles. Les auteurs de Transyt ont considéré (arbitrairement !) qu'une simulation de deux cycles était suffisante pour obtenir des distributions approximativement stationnaires; ceci

toutefois ne semble pas confirmé par nos exemples numériques. Ce choix dans le dessin de Transyt semble avoir été motivé par le désir de limiter l'ampleur des calculs, car en général la procédure d'optimisation teste un grand nombre de valeurs possibles pour la commande. Si chaque valeur entraînait une longue simulation, le temps de calcul serait prohibitif.

Transyt introduit une seconde façon de limiter la taille des calculs. En effet lorsque on modifie la commande à un seul noeud dans le réseau, le recalcul des distributions stationnaires sur les arcs n'implique en général qu'un faible pourcentage du nombre total d'arcs dans le voisinage du noeud où l'on a modifié la commande. Dans ce cas il est avantageux de restreindre la simulation aux arcs affectés par le changement, qui en général constituent un faible pourcentage du nombre total d'arcs, en procédant de proche en proche; c.à.d. qu'une condition nécessaire pour calculer les nouvelles distributions d'arrivées de flots sur les arcs immédiatement en aval de l'arc I est que la distribution pour l'arc I ait elle même été modifiée. Cette approche, ou simulation locale semble induire un ordre dans lequel on doit considérer les arcs du réseau et pose un problème quand les arcs du réseau forment des circuits. La "solution" de Transyt consiste à couper les circuits par le remplacement de certains arcs par des arcs fictifs. Cette façon de procéder, toujours présente dans Transyt-8 bien qu'automatisée, affecte évidemment la précision de la simulation. Nous proposons dans Micro-Transyt une approche plus exacte et plus simple. La présentation de cette solution sera plus claire si au préalable nous décrivons certains aspects des modèles de simulation considérés.

a) Espace de mémoire requis par arc.

Dans Transyt et, pour le moment également dans Micro-Transyt aussi, on associe à chaque arc I du réseau la distribution Flot(I,Step), Step = 1,...,CYCLE, des arrivées (véh./sec.) de véhicules à l'extrémité de sortie de l'arc. Ceci permet de calculer les flots de sortie de l'arc I pour toute la durée du cycle, avant de passer aux calculs

impliquant d'autres arcs du réseau. Le modèle de simulation locale n'a donc plus la forme d'un système dynamique d'état; l'espace de mémoire requis par le modèle en est considérablement accru. En effet, la valeur CYCLE est en général très supérieure au nombre de blocs sur un arc. A titre d'exemple, la valeur maximale de CYCLE acceptée par Transyt-8 et utilisée dans les exemples de Robertson (2) est CYCLE = 60; la description des flots requiert donc 60 positions de mémoire pour chaque arc du réseau !. Nous pensons pouvoir proposer dans un proche avenir un modèle de simulation locale aussi parcimonieux de l'espace de mémoire par arc que le modèle de simulation globale. Toutefois pour l'instant, et comme solution temporaire, nous utilisons les distributions Flot(I,Step), Step = 1,...,CYCLE dans Micro-Transyt. Pour traiter le cas de réseaux assez grands sur un micro-ordinateur il est également possible de limiter la simulation locale à un sous-réseau d'arcs et de noeuds dans le voisinage du noeud N où l'on a modifié la commande car les arcs éloignés du noeud N sont peu susceptibles de voir leurs distributions stationnaires changer significativement.

b) Principe du découplage "amont-aval".

Un élément important à remarquer dans les modèles de simulation de flots proposés dans cet article est la vitesse de propagation finie des flots sur les arcs. Cette observation nous permet d'affirmer que le problème des circuits si laborieusement "résolu" dans Transyt disparait lorsque on introduit l'aspect temporel de la simulation. En effect pour obtenir les nouvelles distributions stationnaires, il suffit d'effectuer localement l'équivalent d'une simulation globale sur plusieurs cycles jusqu'à convergence de toutes les distributions. L'algorithme présenté ci-dessous indique une façon simple d'effectuer la simulation locale par application de ce que nous appelons le principe du découplage amont-aval. Selon ce principe, la structure logique d'un modèle de simulation de flots dans un réseau sera considérablement simplifiée si on observe la règle d'effectuer tous les calculs relatifs aux sorties de flots de tous les arcs considérés pour la simulation avant de commencer les calculs servant à

déterminer les entrées de flots sur des arcs.

c) Description de l'algorithme de simulation locale.

Rappelons que cet algorithme est utilisé pour obtenir les nouvelles distributions stationnaires de flots résultant d'un changement de la commande à un noeud N du réseau.

```
InitInSet;
REPEAT
   SimSetIn;
   SimSetOut;
   SetIn := SetOut;
UNTIL Cardinal(SetIn) = 0;
```

Figure 3 : Simulation locale.

Les rôles des sous-procédures mentionnées dans la figure 3 sont les suivants :

InitInSet : détermine l'ensemble "SetIn" des arcs incidents au noeud N.

SimSetIn : calcule pour chaque arc I de SetIn les queues qui vont s'accumuler à l'extrémité de l'arc et les flots qui vont sortir de l'arc pour la durée d'un cycle.

SimSetOut : commence par créer l'ensemble vide "SetOut"; ensuite, pour chaque arc I immédiatement en aval d'un arc de SetIn, détermine la somme des flots NewFlot(I,Step) entrant sur l'arc I au cours de l'intervale de temps (Step*Del, (Step+1)*Del) pour Step = 0,1,...,CYCLE-1; calcule également les ratios:

$$R = \frac{\sum_{Step=1}^{CYCLE} |NewFlot(I,Step) - Flot(I,Step)|}{\sum_{Step=1}^{CYCLE} Flot(I,Step)}$$

en temps qu'indice du changement intervenu dans la distribution des flots sur l'arc I. Si R est inférieur à une valeur de seuil suffisamment petite, on considère négligeable la différence entre la nouvelle et l'ancienne distribution de flots sur l'arc I; autrement dit, la distribution sur l'arc I est jugée stationnaire. Si R excède la valeur de seuil, l'arc I est alors inclus dans l'ensemble SetOut. La procédure SimSetOut termine par le remplacement Flot := NewFlot. La boucle dans l'algorithme de la figure (3) termine quand l'ensemble SetIn est vide, c.à.d. quand toutes les distributions de flots sur les arcs sont stationnaires. C'est alors que l'on peut recalculer l'indice de performance.

Cet algorithme donne à l'utilisateur le contrôle de l'ajustement du modèle à la réalité. Ainsi en début d'optimisation de la commande il n'est pas nécessaire d'évaluer la performance très exactement; une valeur de seuil élevée résultant en un coût de calcul comparable à celui de Transyt suffit. Au fur et à mesure que la solution approche de l'optimum l'utilisateur peut alors réduire la valeur du seuil et obtenir des distributions réellement stationnaires et par conséquent une évaluation plus précise de la performance associée à la commande.

III - Stratégie d'optimisation de la commande

La stratégie d'optimisation de la commande adoptée par Micro-Transyt est très similaire à celle de Transyt. Dans le cas de Transyt on poursuit les opérations d'optimisation à un noeud jusqu'à ce que les valeurs des paramètres de commandes à ce noeud ne puissent plus être améliorées, avant de passer au noeud suivant. Dans le cas de Micro-Transyt on cherche à améliorer la valeur d'un paramètre seulement marginalement avant de passer au paramètre suivant, puis au noeud suivant,...etc. Il semblerait que cette dernière approche soit moins susceptible de converger sur des optima locaux peu désirables. Cette

stratégie peut être considérée comme une approximation des méthodes de gradient, en général beaucoup plus robustes que les méthodes de relaxation.

IV - Sortie des résultats

Les principales procédures utilisées actuellement pour la sortie des résultats permettent,

1) l'affichage ou impression des valeurs de la commande optimisée ainsi que celui de la valeur de l'indice de performance correspondant,

2) l'affichage ou impression des distributions stationnaires d'arrivées de flots de véhicules sur les arcs,

3) l'animation graphique sur écran Tekronix montrant l'évolution temporelle des flots de véhicules sur le sous-réseau définis à partir d'une paire quelconque de noeuds adjacents du réseau étudié et des arcs qui leur sont incidents.

Conclusion

Nous avons mis en place l'environnement Micro-Transyt sur un micro-ordinateur Ohio-Scientific C8-P analogue à un Apple II avec 64K de mémoire et compilateur UCSD pascal. Ces procédures après quelques ajustements mineurs sont également disponibles sur le système Cyber 170 de l'Université de Montréal. A présent, des essais sur le terrain sont nécessaire pour évaluer l'ampleur des gains de performance attendus de nos modifications et extensions de Transyt.

Les extensions anticipées de ce logiciel comptent, parmi d'autres, l'obtention de meilleures estimations des paramètres utilisés dans les modèles de simulation, la généralisation des types de commandes permises aux noeuds, l'ajustement automatique des coefficients de tourne pour refléter le comportement des conducteurs de véhicules en

réponse à la commande proposée.

A son stade de dévelopement actuel Micro-Transyt peut déjà faciliter l'acquisition et l'utilisation d'un logiciel sophistiqué de régulation des feux de circulation par les personnes responsables de la circulation dans le contexte d'une ville de petite ou moyenne importance.

References

1 - Robertson D.I. (1969) TRANSYT - A Traffic Study Tool. RRL Report LR 253.

2 - TRRL (1981) TRANSYT 8 Users Manual.

3 - Kaltenbach M. A State Space Simulation Model for Urban Traffic Control. Rapport soumis pour publication, Centre de Recherche sur les Transports, Université de Montréal.